
django-oscar-telegram-bot

Documentation

Release 0.5.0

Juan Madurga

October 23, 2016

1	django-oscar-telegram-bot	3
1.1	Documentation	3
1.2	Quickstart	3
1.3	Features	4
1.4	Running Tests	4
2	Installation	5
3	Usage	7
3.1	Screenshots from demo	7
4	Contributing	15
4.1	Types of Contributions	15
4.2	Get Started!	16
4.3	Pull Request Guidelines	16
4.4	Tips	17
5	Credits	19
5.1	Development Lead	19
5.2	Contributors	19
6	History	21
6.1	0.1.0 (2016-01-26)	21

Contents:

django-oscar-telegram-bot

CI: PyPI: Docs: Telegram Bot for Django-Oscar ecommerce.

1.1 Documentation

The full documentation is at <https://django-oscar-telegram-bot.readthedocs.org>.

This package uses django-telegram-bot, take a look to <https://github.com/jlmadurga/django-telegram-bot>

1.2 Quickstart

Install django-oscar-telegram-bot:

```
pip install django-oscar-telegram-bot
```

Add it to your INSTALLED_APPS:

```
INSTALLED_APPS = [
    ...
    'oscar_telegrambot', # this app
    'telegrambot', # django-telegram-bot
    'rest_framework', # django-telegram-bot uses django-rest-framework
    ...
]
```

Configure in settings:

```
TELEGRAM_BOT_COMMANDS_CONF = "oscar_telegrambot.commands"
```

As other django-telegram-bot app set your Bots with admin panel.

Add url to have webhook:

```
url(r'^telegrambot/', include('telegrambot.urls', namespace="telegrambot")),
```

To set the webhook for telegram you need `django.contrib.sites` installed, `SITE_ID` configured in settings and with it correct value in the DB.

You can take a look to a demo repo using the sandbox in <https://github.com/jlmadurga/django-oscar-telegram-bot-demo>. This demo is already installed in <http://django-oscar-telegram-bot-demo.herokuapp.com/en-gb/> and you can test the bot https://telegram.me/djangooscartelegrambotdemo_bot

1.3 Features

- Wellcome and help commands
- Unknown command handler
- Categories list command
- Products list command
- Product detail command
- Order list&detail command only when logged

1.4 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install -r requirements-test.txt
(myenv) $ python runtests.py
```

Installation

At the command line:

```
$ pip install django-oscar-telegram-bot
```

Add it to your `INSTALLED_APPS`:

```
INSTALLED_APPS = [  
    ...  
    'oscar_telegrambot', # this app  
    'telegrambot', # django-telegram-bot  
    'rest_framework', # django-telegram-bot uses django-rest-framework  
    ...  
]
```

Usage

Configure in settings:

```
TELEGRAM_BOT_COMMANDS_CONF = "oscar_telegrambot.commands"
```

As other django-telegram-bot app set your Bots with admin panel.

Add url to have webhook:

```
url(r'^telegrambot/', include('telegrambot.urls')),
```

To set the webhook for telegram you need 'django.contrib.sites' installed, SITE_ID configured in settings and with it correct value in the DB.

You can take a look to a demo repo using the sandbox in <https://github.com/jlmadurga/django-oscar-telegram-bot-demo>. This demo is already installed in <http://django-oscar-telegram-bot-demo.herokuapp.com/en-gb/> and you can test the bot https://telegram.me/djangooscartelegrambotdemo_bot

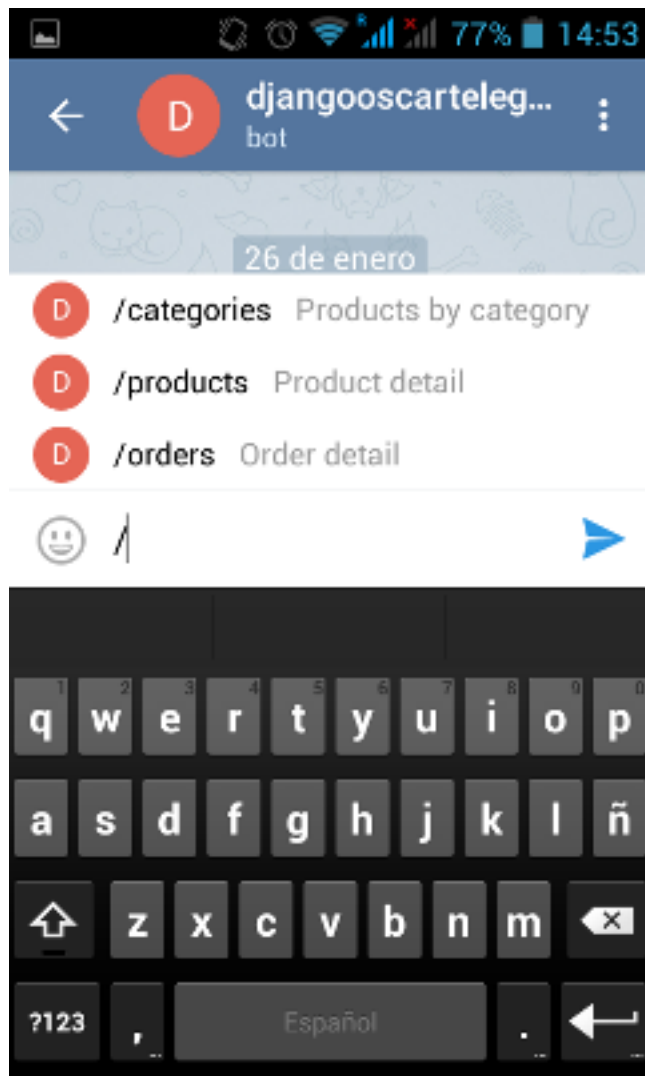
You can override templates of the messages. Just add new templates for texts and keyboards in `your_template_dir/telegrambot/messages/`

3.1 Screenshots from demo

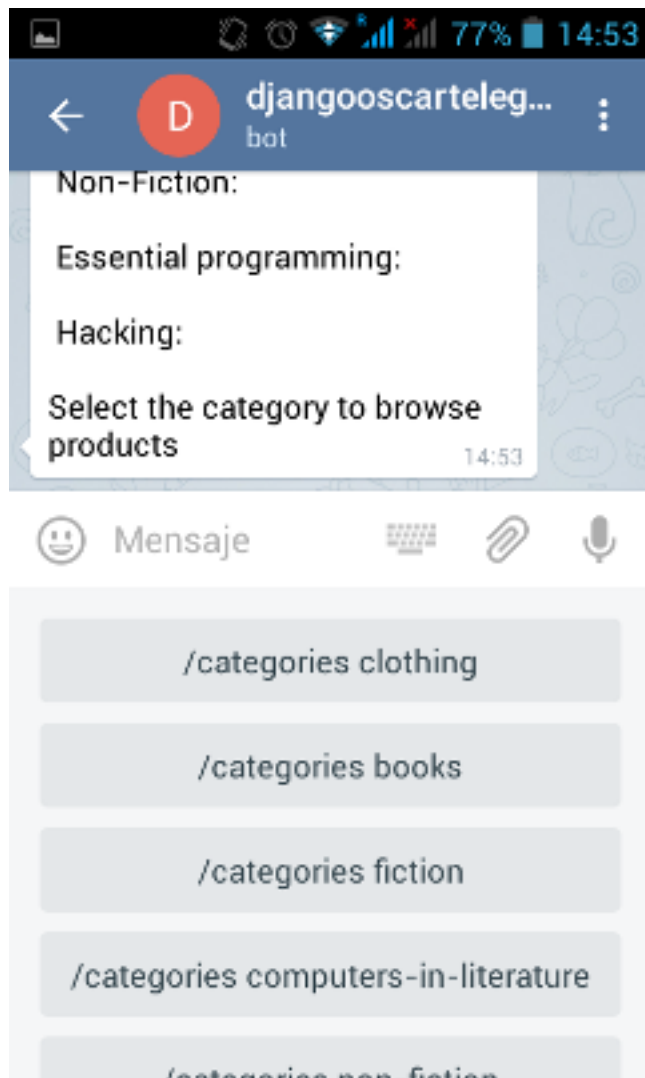
Wellcome:



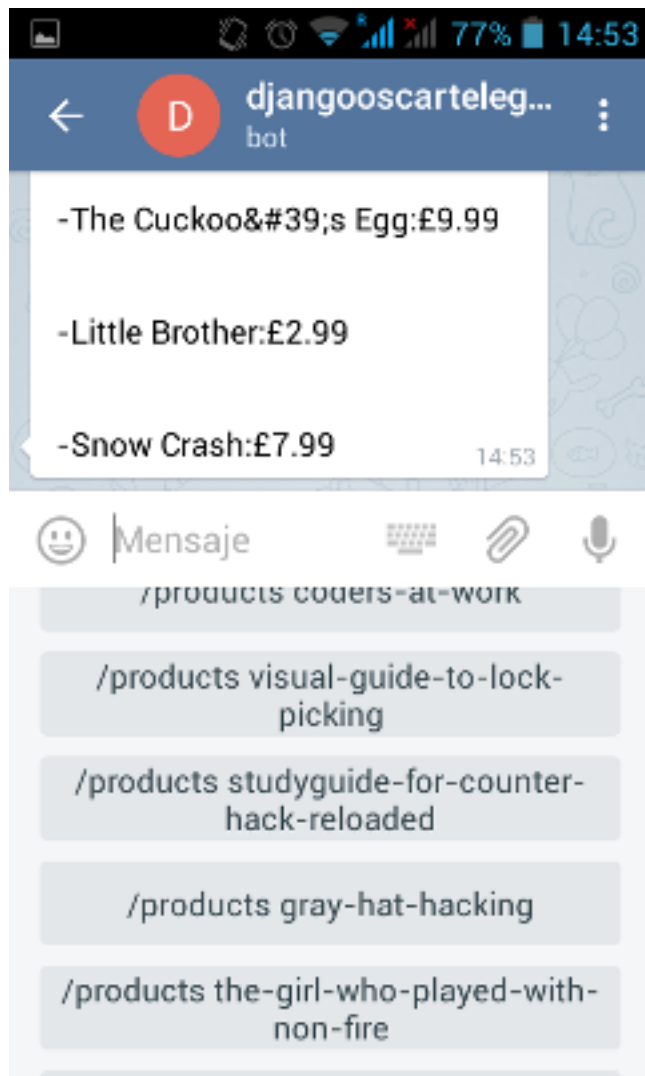
List of commands:



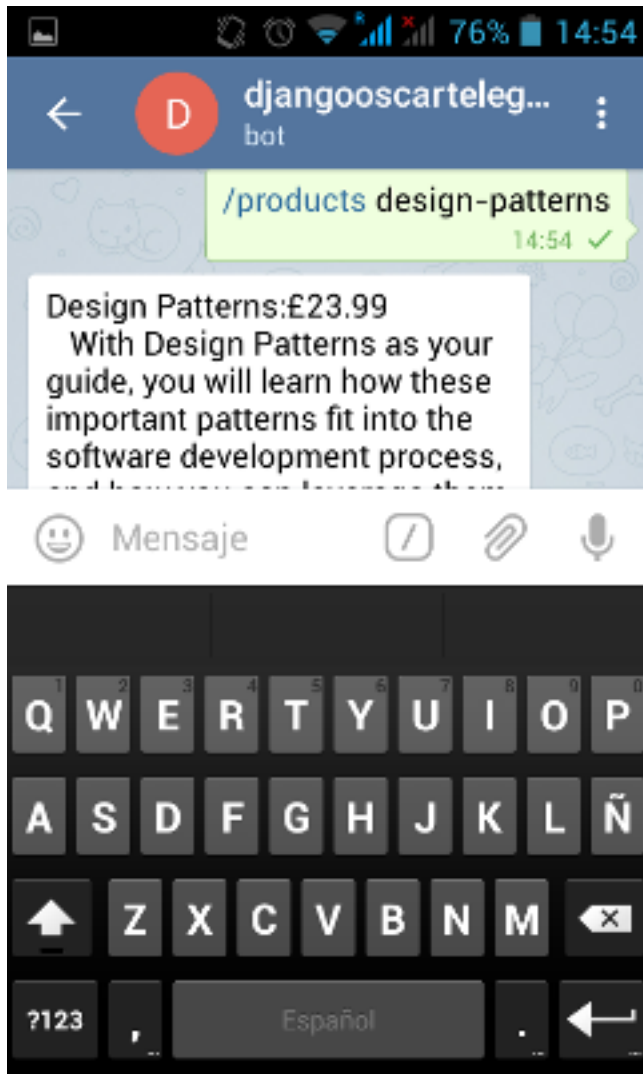
List of categories and keyboard to select the category:



Products of a category. Select the product in keyboard:



Product detail:



Order detail providing order number:



Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/jlmadurga/django-oscar-telegram-bot/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

django-oscar-telegram-bot could always use more documentation, whether as part of the official django-oscar-telegram-bot docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/jlmadurga/django-oscar-telegram-bot/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *django-oscar-telegram-bot* for local development.

1. Fork the *django-oscar-telegram-bot* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-oscar-telegram-bot.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv django-oscar-telegram-bot
$ cd django-oscar-telegram-bot/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 oscar_telegrambot tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/jlmadurga/django-oscar-telegram-bot/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_telegram_bot
```

Credits

5.1 Development Lead

- Juan Madurga <jlmadurga@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.1.0 (2016-01-26)

- First release on PyPI.